

ElasticNFV: An Elasticity Manager for NFV using SDN

A. Heideker, *Member, IEEE*, and C. Kamienski, *Senior Member, IEEE*

Abstract—Information is becoming increasingly ubiquitous in our society and the underlying network infrastructure must provide a flexible and efficient way to cope with ever growing demands. The challenge to provide telecom infrastructure with high quality of service levels, competitive cost and low environmental impact requires new technologies and approaches. SDN and NFV technologies emerge as a solution to these demands, shown new ways to implement dynamic and elastic network services. This paper introduces ElasticNFV, a SDN-based NFV orchestrator capable of performing load balancing in an application oblivious way. Performance analysis results confirm the potential of ElasticNFV to adequately address current telecom infrastructure needs.

Index Terms—IaaS, NFV, SDN, Elasticity.

I. INTRODUÇÃO

O Movimento de migração dos recursos computacionais das empresas, organizações governamentais e não governamentais, institutos de pesquisa e dados pessoais impulsionou a tecnologia de computação em nuvem. Este movimento que começa com o conceito de *Infrastructure as a Service* (IaaS), hoje defende o abrangente conceito de *Everything as a Service* (EaaS)[1], em um ambiente onde os recursos são abundantes, mas finitos, e as responsabilidades sociais e ambientais estão em primeiro plano. Neste cenário, a infraestrutura física de telecomunicações passa a exercer o papel de substrato [2], exigindo novas soluções de gerenciamento destes recursos. Exigindo cada vez mais novas técnicas de gerenciamento, as Cidades Inteligentes, Internet das Coisas (IoT) e as demandas para implantação das redes celulares 5G colocam ainda mais pressão sobre a necessidade de novas abordagens para este problema.

No sentido de atender estas demandas, a tecnologia de virtualização desempenha o papel central, não só tornando possível o conceito de computação em nuvem, mas também inspirando novas abordagens e soluções de rede, como é o caso das Redes Definidas por Software (SDN) e da Virtualização de Funções de Rede (NFV). Apesar destas novas tecnologias representarem soluções eficientes em suas respectivas aplicações, não há consenso sobre as plataformas, linguagens e protocolos para a ampla adoção destas pelo mercado, o qual vem propondo soluções proprietárias e direcionadas à nichos específicos de mercado, como 5G. Este movimento da indústria de telecomunicações representa uma ameaça ao desenvolvimento livre destas tecnologias, indo na contramão de seus fundamentos.

O desafio de equacionar estas demandas, gerenciando dinamicamente os recursos[3], possibilitando não só absorver este crescimento da infraestrutura de telecomunicações como também garantir a qualidade de serviço, baixo custo e impacto ambiental é tratado neste trabalho com a apresentação do ElasticNFV, um gerenciador de elasticidade que utiliza NFV e SDN para instanciar novas funções de rede e direcionar o tráfego em para atender a demanda, dispensando os recursos subutilizados em períodos de ociosidade. O objetivo do ElasticNFV é fornecer um serviço de infraestrutura elástica desacoplada da aplicação, ou seja, transparente fim-a-fim. Os resultados obtidos na avaliação de desempenho apresentada comprovam o funcionamento eficaz do ElasticNFV, mostrando a viabilidade da adoção gerenciador de elasticidade, além de comparar os resultados obtidos com uma solução sem infraestrutura dinâmica.

Na sequência do artigo são apresentadas as tecnologias trabalhos correlatos. A seção III explora a arquitetura do ElasticNFV. A seção IV apresenta a metodologia utilizada na avaliação de desempenho. A seção V apresenta os resultados obtidos na avaliação de desempenho e finalmente na seção VI são apresentadas as discussões a respeito dos experimentos.

II. TECNOLOGIAS E TRABALHOS RELACIONADOS

As *Software Defined Networks* (SDN) [4] representam uma abordagem inovadora na área de redes de computadores, propondo separar os planos de controle e de dados, centralizando a inteligência e o estado da rede, abstraindo a infraestrutura de rede das aplicações, onde o hardware de rede passa a ser considerado apenas o meio físico onde a informação flui, transferindo as decisões sobre esses fluxos de dados para a figura do controlador.

Além dos enlaces envolvidos no trajeto percorrido pela informação, a comunicação fim-a-fim conta ainda com elementos intermediários, responsáveis por tratar estes fluxos de dados, como *Firewalls*, VPN, NAT, entre outros. Estes *middleboxes* ou Funções de Rede, via de regra são implementados por equipamentos dedicados, com hardware e software proprietários e de alto custo, conhecidos como *appliances*. De acordo com o ETSI [5], além do alto custo de aquisição desses equipamentos, as tecnologias utilizadas impedem a evolução e/ou modificação destes equipamentos para implementar novas ideias e tecnologias experimentais. Do ponto de vista do gerenciamento, por vezes, é necessária a presença do profissional no local, via console, além de exigir modificações nas conexões físicas em certas circunstâncias. A tecnologia *Network Functions Virtualization* (NFV), proposta pelo *European Telecommunications Standards Institute* (ETSI), propõe

A. Heideker, Universidade Federal do ABC (UFABC), Santo André, São Paulo, Brasil (e-mail: alexandre.heideker@ufabc.edu.br)

C. A. Kamienski, Universidade Universidade Federal do ABC (UFABC), Santo André, São Paulo, Brasil (e-mail: cak@ufabc.edu.br)

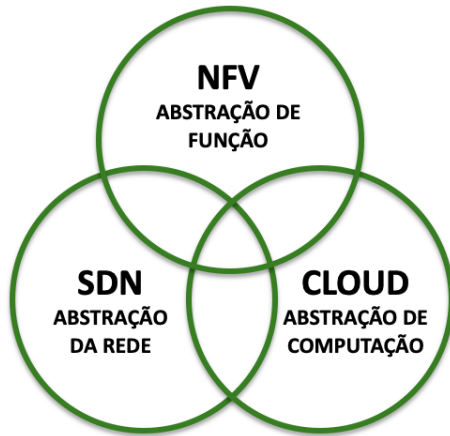


Fig. 1. Relacionamento entre NFV, SDN e Computação em Nuvem (Adaptado de [3]).

a substituição destes equipamentos por versões virtualizadas, executadas sob demanda em servidores de virtualização de prateleira, assim como as máquinas virtuais utilizadas em nuvens públicas e privadas.

A Fig.1 mostra o benefício mútuo presente na intersecção entre NFV e SDN, promovendo a automação, isolamento e agilidade entre NFV e Computação em Nuvem, na orquestração, elasticidade e provimento de recursos. A intersecção das três tecnologias permite a implementação do conceito de IaaS de forma ampla e eficiente.

Wang et al[6] apresentam um problema similar ao que trata este trabalho, porém de forma estática, enfrentando o problema de infraestrutura elástica em uma rede corporativa submetendo os requisitos do *service chaining* (serviço encadeado) ao algoritmo para obter uma espécie de pacote com a configuração necessária para atendê-los. Esta abordagem mostra-se adequada a ambientes corporativos onde os requisitos possuem características estáveis e previsíveis.

Em [7], os autores apresentam a implementação de um sistema de balanceamento de carga utilizando SDN como alternativa ao tradicional uso de um servidor que recebe e encaminha o tráfego para o servidor que responderá à solicitação. Diferente do ElasticNFV, o trabalho de Shang et al é direcionado à aplicação e não à infraestrutura.

Já no trabalho de [8], o problema de balanceamento de caminhos é tratado utilizando OpenFlow, onde os autores realizam uma avaliação de desempenho em simulações e em experimentos reais em pequena escala, utilizando também o controlador Floodlight. Além do objeto de estudo do trabalho ser a sobrecarga de enlaces, assim como em [6], o trabalho considera uma configuração estática da rede, apenas com variação no tráfego submetido aos enlaces que serão balanceados.

Este trabalho apresenta a continuidade dos avanços obtidos em [9], onde o mesmo cenário serve de avaliação de desempenho de diferentes técnicas de virtualização e identifica suas respectivas influências em sua adoção pela tecnologia NFV. O sistema de gerenciamento de elasticidade apresentado em [9] não utiliza a tecnologia SDN, essencial para desacoplar a solução do cenário proposto, exigindo interferência direta do

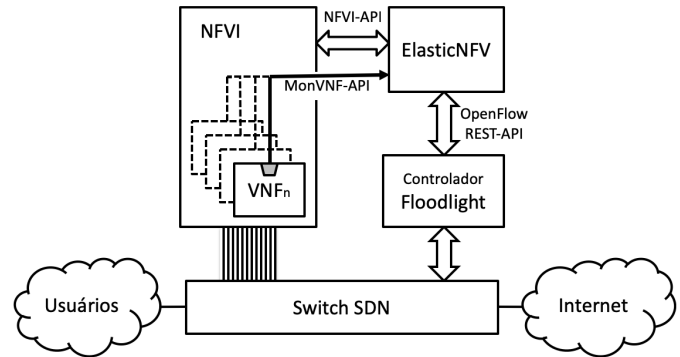


Fig. 2. Arquitetura do ElasticNFV.

gerenciador de elasticidade na infraestrutura da praça digital.

III. GERENCIADOR DE INFRAESTRUTURA ELÁSTICA: ELASTICNFV

A Fig.2 apresenta o diagrama da arquitetura do ElasticNFV, onde é observada a presença do *switch* SDN, responsável por realizar o enlace entre os usuários e a respectiva VNF responsável pelo seu tráfego, além da modificação dos cabeçalhos dos respectivos pacotes. Tem-se a figura do *Network Function Virtualization Infrastructure* (NFVI), responsável por instanciar e alocar as Funções de Rede Virtualizadas (VNF). Ainda na Fig.2 tem-se o controlador SDN, papel este desempenhado pelo Floodlight e finalmente o ElasticNFV, responsável pelo gerenciamento e orquestração da solução de elasticidade.

A. NFVI

De acordo com a arquitetura MANO apresentada pelo ETSI [ETSI et al. 2012], a *Network Functions Virtualisation Infrastructure* (NFVI) é responsável por fornecer a infraestrutura de virtualização necessária para implementar as funções de rede, ou seja, as máquinas virtuais e seus enlaces para prover o serviço. Em [10] esta infraestrutura é fornecida pelo OpenStack[11], controlador de nuvem computacional amplamente utilizado. Diversas técnicas de virtualização podem ser utilizadas para fornecer estas máquinas virtuais, e as influências destas técnicas são abordadas em [9]. Neste trabalho foi utilizada a virtualização por container[12], porém, qualquer tipo de virtualização poderia ser utilizada incluindo cenários onde diversos tipos de máquinas virtuais coexistem.

B. MonVNF - Monitor de VNF

Entre os muitos desafios explorados na área de NFV, a detecção de gargalos, ou seja, sobrecarga de uma VNF, é fundamental para um sistema de elasticidade de infraestrutura desacoplado da aplicação, como é o caso do ElasticNFV. Apesar da solução de virtualização fornecida pela NFVI ser indiferente, no caso da detecção da sobrecarga de uma VNF, tanto o tipo de serviço virtual fornecido (NAT, *Proxy*, *Firewall*, etc.) como a tecnologia de virtualização empregada influenciam esta métrica, sendo necessário, em diversas situações, a adoção não de uma, mas de um conjunto de métricas

TABELA I
MÉTRICAS FORNECIDAS PELO MONVNF

Métrica	Descrição
CPU	Identifica o percentual de uso do tempo da vCPU (percentual)
Memória	Quantidade de bytes utilizado da memória RAM da máquina virtual (unitária e percentual)
Tráf.Interface 1	Tráfego em bytes na primeira interface de rede desde sua instanciação
Tráf.Interface 2	Tráfego em bytes na segunda interface de rede desde sua instanciação
Level	Métrica mapeada para o tipo de função de rede (percentual)

para tanto. Em [13], os autores apresentam a proposta de padronização da API de monitoramento destas métricas.

Para implementar o ElasticNFV, foi desenvolvido um *daemon* que é instalado no sistema operacional da VNF, fornecendo a qualquer instante um conjunto de métricas que determinam o nível de utilização desta. Além das métricas mais comuns, a personalização deste *daemon* tanto pelo fornecedor da VNF como do provedor de NFVI podem fornecer através da métrica “*level*” a exata medida de utilização dos recursos da VNF. A TABELA I apresenta as métricas disponíveis no MonVNF.

Na avaliação realizada neste trabalho, utilizou-se a métrica do percentual de uso da tabela *Conntrack*. Esta tabela é responsável pelo rastreamento das conexões efetuadas em um *Network Address Translator* (NAT), utilizado nos experimentos. Em experimentos preliminares foi possível observar que tanto a variação de carga de CPU como o uso de memória não apresentam uma correlação com a sobrecarga do NAT, ao contrário da tabela *conntrack*[9]. O tamanho da tabela *conntrack* é definida pelo *kernel* do sistema operacional de acordo com a memória RAM disponível no momento de sua inicialização, utilizando (1).

$$Conntrack = \frac{RAM}{131072} \times 8 \quad (1)$$

Quando o limite da tabela *conntrack* é atingido, novas conexões não são admitidas e antigas conexões são descartadas de forma imprevisível. Apesar de ser possível manipular o tamanho da tabela *conntrack*, esta sempre representa o limite para a quantidade de conexões que um NAT implementado pelo módulo *netfilter* do Linux é capaz de rastrear.

C. Controlador SDN: Floodlight

A tecnologia SDN apresenta a separação do plano de dados do plano de controle, ou seja, o plano de dados torna-se um simples condutor de dados e o controle de como este processo de condução, antes realizado de forma descentralizada nos *switches*, é delegado ao controlador, que pode ser centralizado ou distribuído mas, ao contrário dos antigos *switches*, fornece uma abstração completa da rede. Em 2008, a introdução do OpenFlow [4], alavanca o desenvolvimento de pesquisas em SDN e a sua posterior adoção pelo mercado.

Há diversos controladores disponíveis no mercado e entre eles o Floodlight[14] foi escolhido para os experimentos. Apesar de possuir uma rica API em Java, o ElasticNFV

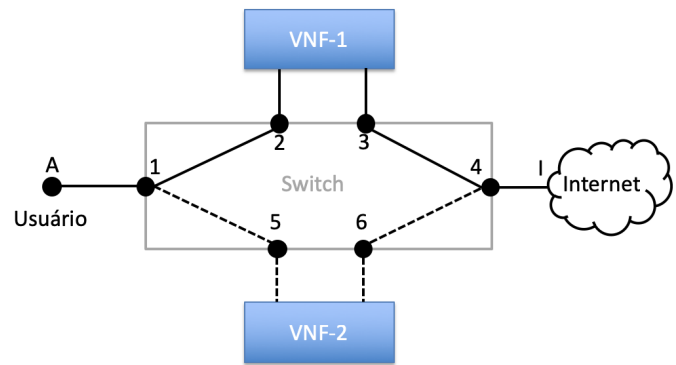


Fig. 3. Grafo de encaminhamento.

utiliza a interface REST para interagir com o controlador, desacoplando a solução do Floodlight para permitir o uso de outros controladores que forneçam a interface REST. Uma característica essencial, tanto do controlador como do *switch* utilizado pelo ElasticNFV é o suporte ao OpenFlow 1.1[15].

D. Grafo de Encaminhamento

Com as informações fornecidas pela NFVI a respeito das VNFs instanciadas, o ElasticNFV realiza uma consulta ao controlador para que o mesmo forneça a informação sobre os dispositivos conectados na rede física. De posse destas informações, o ElasticNFV constrói um grafo de encaminhamento. A Fig.3 apresenta um exemplo de grafo onde o caminho A-1-2 e 3-4-I definem o uso da VNF-1. Durante o processo de balanceamento de carga, é necessário definir um novo caminho para o usuário A, em A-1-5 e 6-4-I. O ElasticNFV determina este grafo através do mapeamento entre endereço de hardware *ethernet* (MAC) e o número da porta do *switch*.

Diante deste mapeamento, são criadas regras de fluxo estático onde o tráfego oriundo de um determinado MAC é encaminhado para outra porta diferente do caminho original, além da reescrita dos cabeçalhos TCP(endereço MAC e IP de destino). A comunicação no sentido oposto também é modificada por este mecanismo para que o usuário não perceba a manipulação do tráfego. Em tempo de execução, o ElasticNFV monitora a presença de novos dispositivos na rede além da sobrecarga das VNFs, quando o mecanismo de elasticidade atua solicitando a criação de uma nova VNF, elaborando as regras que serão enviadas ao controlador para o posterior envio ao *switch* e finalmente desviar parte do tráfego para a nova VNF.

E. Algoritmo de Elasticidade

Apenas para prova de conceito foi implementado um algoritmo simples de elasticidade para o ElasticNFV, porém, sua estrutura permite a criação de outros algoritmos utilizando técnicas preditivas ou reativas, tornando a solução ainda mais poderosa. A Fig.4 mostra este algoritmo que utiliza o *LimiarSuperior* e o *LimiarInferior* como parâmetros principais em sua execução.

```

1  Início:
2  LuzVermelha<-0; LuzVerde<-0;
3  Looping {
4    atualizaListaDispositivosRastreados;
5    AtualizaEstadoDasVNFs;
6    Se (CargaMedia > LimiarSuperior)
7      LuzVermelha++;
8    senão
9      LuzVermelha<-0;
10   Se (CargaMedia < LimiarInferior)
11     LuzVerde++;
12   senão
13     LuzVerde<-0;
14   Se (LuzVermelha>Tolerancia)
15     CriaNovaVNF;
16     LuzVermelha<-0;
17   Se (LuzVerde>Tolerancia)
18     TransfereTrafegoVNFMenosUtilizada;
19     DestroiVNFMenosUtilizada;
20     LuzVerde<-0;
21   RedistribuiTrafegoEntreVNFsAtivas;
22 }

```

Fig. 4. Algoritmo de Elasticidade.

Na linha 2 do algoritmo as variáveis *LuzVermelha* e *LuzVerde* são inicializadas com o valor zero – estas variáveis são utilizadas para evitar que o algoritmo responda à picos instantâneos de sobre carga ou sub-utilização, permitindo a acomodação do tráfego. O parâmetro *Tolerancia* indica quantos ciclos do algoritmo serão tolerados neste picos. Na linha 4, o algoritmo se comunica com o controlador para obter a lista de dispositivos atualmente rastreados. Na linha 5 os níveis de utilização e demais métricas são coletados pelo ElasticNFV através do MonVNF. Caso o nível médio das VNFs esteja acima do limiar superior, na linha 7 a variável *LuzVermelha* é incrementada – caso contrário ela é zerada. O mesmo acontece no caso do nível médio estar abaixo do limiar inferior, quando na linha 11 a variável *LuzVerde* é incrementada.

Caso a variável *LuzVermelha* ultrapasse o valor do parâmetro *Tolerancia*, na linha 15 uma nova VNF é criada. No caso da *LuzVerde* ultrapassar esta tolerância, o algoritmo transfere, na linha 18, o tráfego da VNF com menor nível de utilização para as outras VNFs ativas e na linha 19 destrói esta VNF. Finalmente na linha 21 os dispositivos rastreados são redistribuídos pelas VNFs ativas para balancear a carga entre elas.

IV. AVALIAÇÃO DE DESEMPENHO

O cenário de cidades inteligentes, onde o fornecimento público de internet ao cidadão, em larga escala, é um exemplo claro onde grandes investimentos são necessários para uma demanda heterogênea e sazonal. Grandes concentrações de pessoas como em eventos esportivos, culturais e sociais podem gerar de forma instantânea e imprevisível uma grande demanda por recursos de telecomunicação. Existem locais onde a demanda pode cair a zero em horários específicos e a níveis altíssimos em determinadas épocas do ano. Fornecer este serviço com a abrangência desejada, com a qualidade de serviço adequada e com a tecnologia convencional é proibitivo do ponto de vista financeiro e energético.

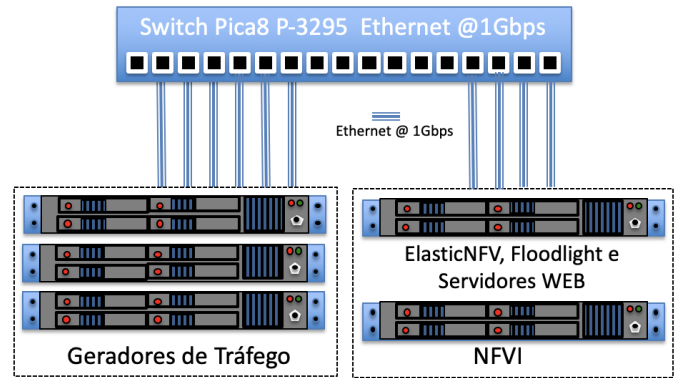


Fig. 5. Esquema físico do experimento.

A partir de 2014, a Prefeitura de São Paulo deu início ao projeto WiFi Livre SP[16], com o objetivo de fornecer acesso público à internet em 120 praças da cidade [17], utilizando tecnologias e topologias tradicionais de rede. Além de questões socioculturais apresentadas por esta iniciativa, a qualidade do serviço oferecido também representa um grande desafio neste projeto, apresentando um cenário perfeito para avaliar o desempenho do ElasticNFV em uma situação onde recursos providos de forma dinâmica garantem a qualidade de serviço e a viabilidade financeira do projeto.

A. Metodologia

Para representar o cenário proposto do experimento foi implementado o esquema mostrado na Fig.5 onde três servidores Intel® Xeon® E3-1240V2, 3.40GHz, 4 núcleos físicos, são responsáveis por gerar o tráfego típico de cada uma das praças do projeto WiFi Livre SP, através de 117 máquinas virtuais (apesar do projeto WiFi Livre SP contar com 120 praças, os dados de apenas 117 praças estavam disponíveis). O experimento ainda conta com um servidor com a mesma configuração dedicado ao fornecimento das máquinas virtuais (NFVI) e um último servidor que hospeda o controlador Floodlight, o ElasticNFV e os servidores web responsáveis por receber as solicitações dos usuários das praças.

Todos os servidores possuem uma segunda conexão de rede utilizada para tráfego de controle e monitoramento, evitando assim influir nos resultados dos experimentos. A Fig.6 mostra o esquema lógico do experimento, com as máquinas virtuais geradoras de tráfego configuradas nos endereços ip 192.168.1.x, 192.168.2.x e 192.168.3.x. As VNFs NAT são instanciadas com três interfaces de rede: uma na sub-rede das praças (192.168.0.x), a segunda na sub-rede dos servidores web (10.0.0.x) e a terceira na sub-rede de controle (10.1.1.X). As conexões entre as máquinas virtuais e o *switch* físico são feitas através do Open vSwitch[18], um *switch* virtual amplamente utilizado e que possui suporte a OpenFlow 1.1.

Para modelar o tráfego foi desenvolvido um gerador de tráfego em linguagem C que realiza requisições de transferência de arquivos em intervalos seguindo uma distribuição exponencial. O tempo médio entre as requisições foi ajustado individualmente em cada máquina virtual para que o número médio de requisições representasse adequadamente o tráfego

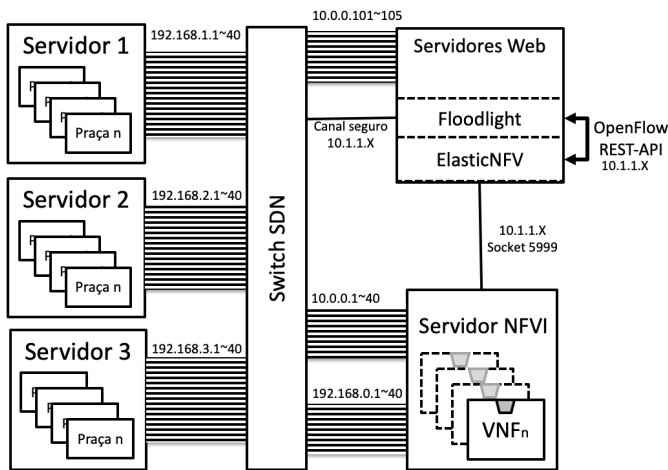


Fig. 6. Esquema lógico do experimento.

gerado pela respectiva praça. Os arquivos requisitados pelo gerador de tráfego são selecionados uniformemente, com seus tamanhos seguindo uma distribuição lognormal, resultando desta forma em um tráfego com distribuição lognormal. Além de realizar a transferência do arquivo selecionado, a cada dez segundos, cada cliente dispara um pacote ICMP (*ping*) até o servidor web para determinar o *Roud Trip Time* (RTT).

Durante os experimentos o ElasticNFV registra, a cada dez segundos, as métricas de uso das VNFs, o número de regras OpenFlow utilizadas, o número de VNFs utilizadas e finalmente o número de dispositivos rastreados pelo controlador SDN. O gerador de tráfego efetua o registro da vazão obtida em cada transferência de arquivo e o RTT até o servidor web. Para obter um intervalo de confiança de 99%, foram realizadas 30 repetições em cada experimento.

V. RESULTADOS

A Fig.7 apresenta a série temporal de uma das repetições efetuadas para os parâmetros de limiar superior em 60% e de inferior em 30%. A figura mostra o número de conexões crescente até aproximadamente um terço do tempo do experimento, quando a segunda VNF é instanciada. Como a carga média observada está acima dos 60%, uma terceira VNF é instanciada permanecendo ativa até a terceira parte do experimento quando a carga média cai abaixo do limiar inferior de 30%, destruindo as instancias da terceira e segunda VNF.

A Fig.8 apresenta o resultado da avaliação do nível médio de utilização das VNFs durante todo o experimento e apenas durante o período de máxima utilização (*link cheio*). Como esperado, o nível de utilização está relacionado com o limiar superior.

A mesma avaliação agora considerando a quantidade média de VNFs utilizadas para suprir a demanda é exibida na Fig.9. Esta métrica está relacionada diretamente com o custo de operação do sistema, já que o modelo de negócios tanto de computação em nuvem como de NFV prevê a remuneração por tempo de uso de máquina virtual. Os resultados mais uma vez mostram uma correlação com o limiar superior do ElasticNFV.

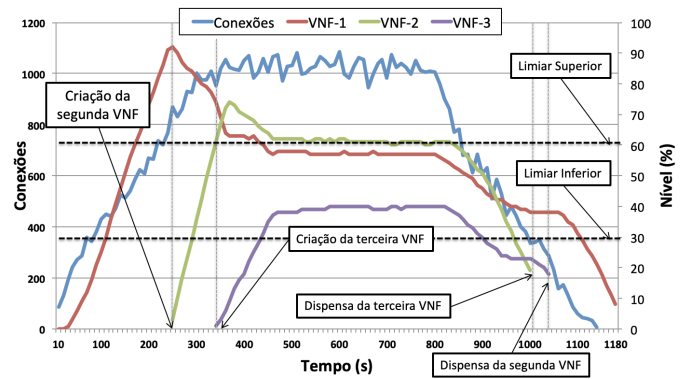


Fig. 7. Série temporal de um dos experimentos com limiar superior de 60% e limiar inferior de 30%.

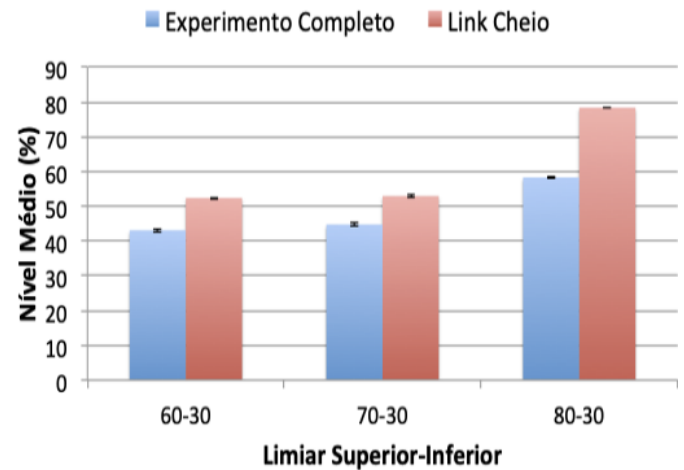


Fig. 8. Nível médio de utilização das VNFs.

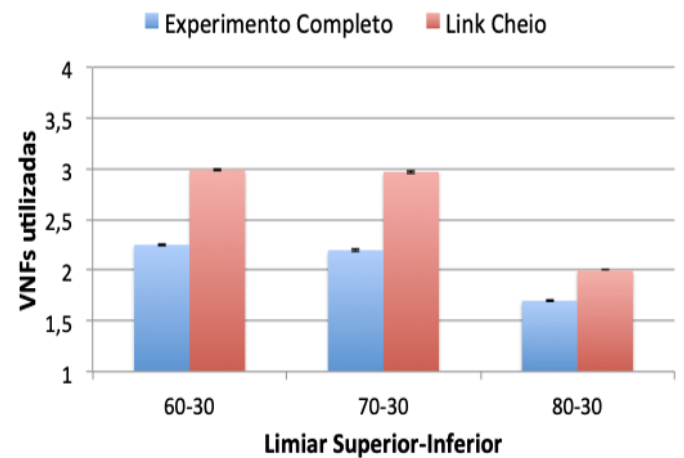


Fig. 9. Quantidade média de VNFs utilizadas.

A Fig.10 mostra a vazão média observada pelos usuários da praça, considerando o experimento completo e o período de máxima utilização do *link*, apresentando uma redução da vazão durante este período como esperado. Aqui é possível observar que o uso do limiar superior em 80% proporciona um melhor desempenho desta métrica, justificável pelo menor número de VNFs utilizadas e a consequente redução da transferência de

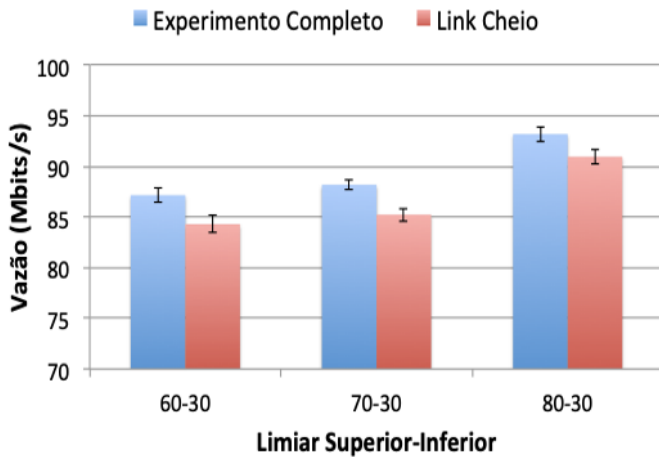


Fig. 10. Vazão média observada pelos usuários.

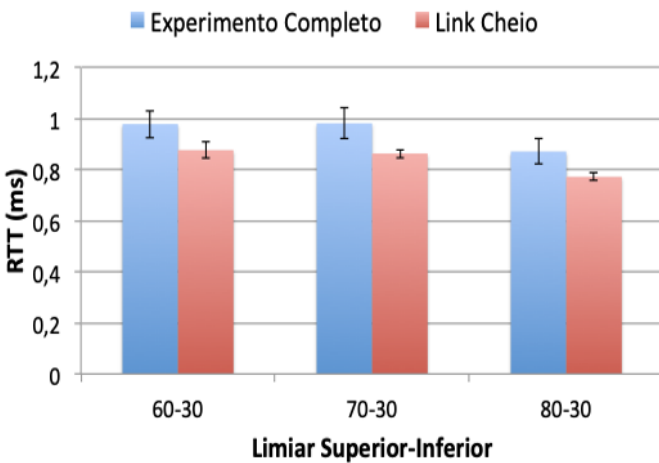


Fig. 11. RTT médio entre o usuário e o servidor web.

tráfego entre diferentes VNFs.

Ainda considerando o ponto de vista do usuário, a métrica de RTT entre o usuário e o servidor web é apresentada na Fig.11. Apesar de sutil, é possível observar uma pequena redução do RTT tanto com a adoção do limiar superior de 80%, como entre a operação com *link* cheio e durante o experimento. A justificativa é a mesma da Fig.10, já que durante o período de máxima utilização não ocorrem desvios no tráfego entre as VNFs – esta troca ocorre apenas quando uma nova VNF é criada ou removida do sistema.

A Fig.12 relaciona o número de conexões simultâneas dos usuários com a vazão observada por estes em relação às diferentes combinações de limiares avaliadas. Mais uma vez os resultados obtidos pelo limiar superior em 80% são superiores na maior parte das concentrações de conexões.

A mesma avaliação dos limiares em relação ao número de conexões simultâneas, agora sobre a métrica do RTT é apresentada na Fig.13, mostrando que esta métrica não é afetada diretamente pela escolha dos limiares, tão pouco pelo número de conexões, considerando que há sobreposição dos intervalos de confiança em todas as variações de parâmetros.

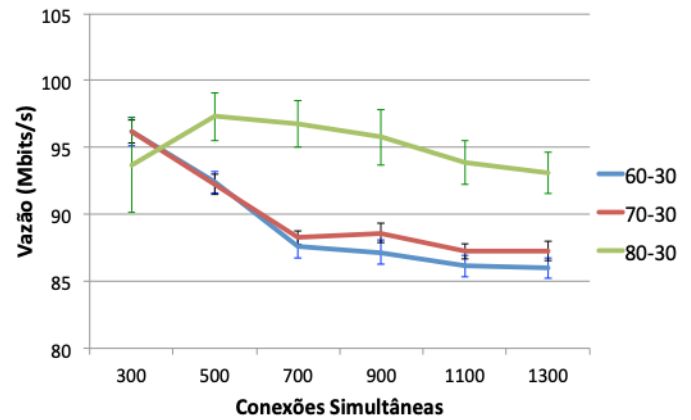


Fig. 12. Vazão média por número de conexões simultâneas.

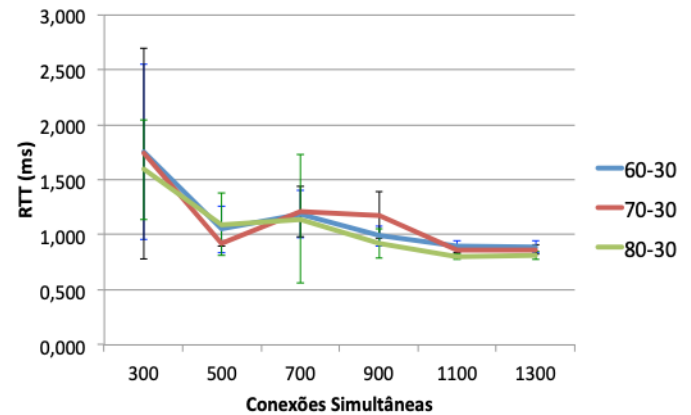


Fig. 13. RTT médio entre o usuário e o servidor web por número de conexões simultâneas.

VI. DISCUSSÕES

O objetivo do desenvolvimento do ElasticNFV, como já denotado, é oferecer uma plataforma de orquestração de VNFs utilizando SDN, fornecendo um comportamento elástico da infraestrutura virtualizada, e, como visto nos resultados obtidos na Fig.9, utilizando de forma mais eficiente os recursos físicos, gerando uma redução no custo de operação (OPEX) e o conseqüente impacto ambiental, mesmo com o uso de um algoritmo de elasticidade bastante simples.

Tanto a vazão (Fig.12) como o RTT (Fig.11) observados nos experimentos apresentam valores médios compatíveis com os limites das tecnologias de enlace utilizadas, e as variações observadas nos resultados são reflexo unicamente da variação dos parâmetros utilizados no algoritmo de elasticidade. Esta observação permite concluir que o ElasticNFV é uma alternativa viável à implementação em ambientes de produção. Outra característica presente no ElasticNFV é a possibilidade de integração transparente com arquiteturas de Computação em Névoa[19], mitigando problemas relacionados à requisitos de latência e utilização de infraestrutura local. Qualitativamente, o uso de uma plataforma baseada em software simplifica significativamente o gerenciamento da infraestrutura, além de permitir a modificação no comportamento da rede de forma centralizada, simplificar o processo de auditoria e garantia

de *Service Level Agreement* (SLA) e o finalmente o uso de algoritmos sofisticados e técnicas analíticas para interação com a rede como abordados em [20].

VII. CONCLUSÃO

Este trabalho apresentou o ElasticNFV, um gerenciador de elasticidade utilizando SDN e NFV para prover uma infraestrutura dinâmica e elástica fim-a-fim, independente da aplicação. Os resultados apresentados demonstram o correto funcionamento do sistema proposto, demonstrando resultados compatíveis com as soluções tradicionais de infraestrutura de telecomunicações, porém com um melhor aproveitamento dos recursos disponíveis, gerando economia na implantação e manutenção desta infraestrutura, garantindo uma redução substancial no impacto ambiental decorrente do crescimento destas redes.

Uma investigação mais cuidadosa deve ser realizada sobre o impacto dos limiares superior e inferior no desempenho do sistema, assim como melhorias no algoritmo de elasticidade para uma resposta mais eficaz, bem como garantir sempre a melhor relação custo benefício para o gestor da rede. A avaliação de diferentes configurações, utilizando diferentes *middleboxes*, assim como o estudo de *service chainings* complexos apresentam oportunidades de pesquisa futura.

REFERÊNCIAS

- [1] P. Banerjee, R. Friedrich, C. Bash, P. Goldsack, B. Huberman, J. Manley, C. Patel, P. Ranganathan, and A. Veitch, "Everything as a service: Powering the new information economy," *Computer*, vol. 44, no. 3, pp. 36–43, 2011.
- [2] R. P. Esteves, L. Z. Granville, and R. Boutaba, "On the management of virtual networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 80–88, 2013.
- [3] R. Mijumbi, J. Serrat, J.-L. Gorricho, S. Latré, M. Charalambides, and D. Lopez, "Management and orchestration challenges in network functions virtualization," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 98–105, 2016.
- [4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
- [5] G. ETSI, "001:" network functions virtualisation (nfv)," *Architectural framework*, 2013.
- [6] X. Wang, C. Wu, F. Le, A. Liu, Z. Li, and F. Lau, "Online vnf scaling in datacenters," in *Cloud Computing (CLOUD), 2016 IEEE 9th International Conference on*. IEEE, 2016, pp. 140–147.
- [7] Z. Shang, W. Chen, Q. Ma, and B. Wu, "Design and implementation of server cluster dynamic load balancing based on openflow," in *Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), 2013 International Joint Conference on*. IEEE, 2013, pp. 691–697.
- [8] S. Govindraj, A. Jayaraman, N. Khanna, and K. R. Prakash, "Openflow: Load balancing in enterprise networks using floodlight controller," *University of Colorado*, 2012.
- [9] A. Heideker and C. Kamienski, "Gerenciamento flexível de infraestrutura de acesso público à internet com nfv," in *XXXIV Simpósio Bras. Redes Comput. e Sist. Distrib.SBRC. SBC.*, 2016.
- [10] —, "Funções de rede virtualizadas em plataforma de computação em nuvem para cidades inteligentes," in *XIII Workshop em Clouds e Aplicações—WCGA. SBC. p43–56*, 2015.
- [11] The openstack project. Accessed: 2017-02-10. [Online]. Available: <https://www.openstack.org>
- [12] L. Foundation. Linux containers lxc. Accessed: 2016-12-20. [Online]. Available: <https://linuxcontainers.org>
- [13] J. Khalid, M. Coatsworth, A. Gember-Jacobson, and A. Akella, "A standardized southbound api for vnf management," in *Proceedings of the 2016 workshop on Hot topics in Middleboxes and Network Function Virtualization*. ACM, 2016, pp. 38–43.

- [14] Project floodlight. Accessed: 2017-03-15. [Online]. Available: <http://www.projectfloodlight.org>
- [15] B. Pfaff, B. Heller, D. Talayco, D. Erickson, G. Gibb, G. Appenzeller, J. Tourrilhes, J. Pettit, K. Yap, M. Casado *et al.*, "Openflow switch specification," 2009.
- [16] P. de São Paulo. Wifilivre. Accessed: 2017-02-10. [Online]. Available: <http://wifilivre.sp.gov.br>
- [17] J. Ratusznei, W. Silva, N. Pinheiro, R. Melo, and C. Kamienski, "Uma rede wifi aberta de larga escala como infraestrutura para cidades inteligentes," *42º SEMISH-Seminário Integrado de Software e Hardware. SBC*, 2015.
- [18] L. Foundation. Open vswitch. Accessed: 2017-02-10. [Online]. Available: <http://openvswitch.org>
- [19] I. Zyrianoff, F. Borelli, G. Biondi, A. Heideker, and C. Kamienski, "Scalability of real-time iot-based applications for smart cities," in *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2018, pp. 00 688–00 693.
- [20] A. Heideker, I. Zyrianoff, and C. A. Kamienski, "Profiling service function chaining behavior for nfv orchestration," in *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2018, pp. 01 020–01 025.



Alexandre Heideker received a B.S. degree in computer science from the Federal University of ABC (Santo André, Brazil) in 2011, a M.S. degree from the Federal University of ABC (Santo André, Brazil) in 2016. He is Ph.D. candidate in Information Engineering at Federal University of ABC (UFABC) in Santo André SP, Brazil, and is a member of the NUVEM Strategic Research Group. His current research interests include network softwarization, Infrastructure as a Service (IaaS) and cloud/fog computing.



Carlos A. Kamienski received a B.S. degree in computer science from the Federal University of Santa Catarina (Florianópolis, Brazil) in 1989, a M.S. degree from the State University of Campinas (Campinas, Brazil) in 1994 and a Ph.D. in computer science from the Federal University of Pernambuco (Recife Brazil) in 2003. He is full professor of computer science at the Federal University of ABC (UFABC) in Santo André SP, Brazil, where currently he also holds the positions of head of international relations and head of the NUVEM Strategic Research Group. His current research interests include Internet of Things, Smart Cities, network softwarization and cloud/fog computing.